UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/826,762	04/16/2004	John Harper	119-0033US	1223

29855   7590   01/30/2008
WONG, CABELLO, LUTSCH, RUTHERFORD & BRUCCULERI,
L.L.P.
20333 SH 249
SUITE 600
HOUSTON, TX 77070

EXAMINER
HSU, JONI

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
01/30/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _October 30, 2007_.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-85_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-85_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some *   c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date _____.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

Response to Arguments

1.   Applicant's arguments, see pages 12-36, filed October 20, 2007, with respect to the

rejection(s) of claim(s) 75 under 35 U.S.C. 102(e) and claims 1-74, 76-78, and 85 under 35

U.S.C. 103(a) have been fully considered and are persuasive. Therefore, the rejection has been

withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view

of Grantham (US006215495B1) and Boudier (US006995765B2).

2.   As per Claim 75, Applicant argues domain of definition, region of interest are directed to

area in image, while Olano (US006717599B1) refers to math reduction of tree structure (p. 12).

        In reply, Examiner agrees. But, new grounds of rejection are made in view of Boudier.

3.   Applicant argues the Examiner has combined an excessive number of references, and the

references have no logical connection to each other (p. 17).

        In reply to argument examiner has combined excessive number of references, reliance on

large number of references in rejection doesn't, without more, weigh against obviousness of

claimed invention. See *In re Gorman*, 933 F.2d 982, 18 USPQ2d 1885 (Fed. Cir. 1991).

References have logical connection to each other since combined references suggest advantages

for incorporating their teachings into main reference, as discussed in 103 rejections below.

4.   As per Claim 1, Applicant argues Olano portion cited discusses estimating size of anti-

aliasing filter; there is no discussion regarding a request for filters between processes (p. 22) or

of applying function of filter to image, yielding pixel-image result (p. 23). As per Claim 22,

Olano does not teach claimed division of work between microprocessors. Olano teaches

programs for doing math, not for making or editing images (p. 24). As per Claim 28, Olano's

optimal transformation is result of editing text file of configuration rules in order to adapt

Olano's compiler, this is not optimization claimed (p. 27-28).

In reply, new ground of rejection is made in view of Grantham to more clearly teach this.

5.   As per Claim 24, Applicant argues API in Sturges (US005854637A) is for co. (p. 20).

In reply, McCrossin teaches API (c. 7, ll. 33-35), Sturges is only used for its teaching of

shared memory.

6.   As per Claim 22, Applicant argues McCrossin does not discuss any data structure. API is

generic program interface, it can not teach specifically claimed data structure (p. 24-26).

In reply, Examiner points out McCrossin teaches transform object 103 has relationship

between initial image and specific filter (c. 6, ll. 51-56), so transform object 103 is data structure.

7.   As per Claim 7, Applicant argues Parikh (US006411301B1) teaches pre-compiling list of

display lists. Pre-compiling doesn't yield a program and doesn't teach image result (p. 33).

In reply, Parikh teaches one microprocessor that emulates graphics processing (c. 26, ll.

29-62). So, Grantham is modified so only single microprocessor is used instead of using separate

graphics processor (c. 5, ll. 2-23) so compiled program of Grantham (c. 7, ll. 50-57) is for single

microprocessor, as suggested by Parikh, so combination of Grantham, Parikh teaches Claim 7.

8.   Applicant's arguments filed October 30, 2007, with respect to Claims 79-84 have been

considered but are not persuasive.

9.   As per Claim 79, Applicant argues McCrossin (US006600840B1) has only read and write

APIs while claim recites APIs to filter, image, context and vector objects (p. 14).

In reply, Examiner points out McCrossin describes "an application calls the

transformation object using a procedure call or API call" (c. 7, ll. 33-34), and "transform object

103 determines an actual image vector 139...image request vector 127...is compared to actual

image vector 139 to determine what filters are needed" (c. 6, ll. 51-56). So, McCrossin does

teach APIs to filter objects and other objects, and does not only comprise of read and write APIs.

10.   Applicant argues that McCrossin does not teach context creation (p. 15).

In reply, according to Applicant's disclosure, context is space such as defined place in

memory in which result of filtering operation resides (p. 9, [0045]). McCrossin teaches buffer

107 in which result of filtering operation resides (c. 5, ll. 59-60; c. 6, ll. 51-56), so teaches

context creation.

11.   Applicant argues single reference to "a procedure call or API call" was found in

McCrossin and inappropriately extrapolated to contend McCrossin teaches claim (p. 15).

In reply, Examiner points out McCrossin explicitly recites "an application calls the

transformation object using a procedure call or API call" (c. 7, ll. 33-35), and this passage is

being interpreted exactly the way it is recited, and so reference is not being misinterpreted.

12.   As per Claim 84, Applicant argues McCrossin teaches retrieving filter objects from pre-

existing library rather than using API to create filter objects (p. 16).

In reply, McCrossin teaches application calls transformation object using API call (c. 7,

ll. 33-35), transform object adds filter to filter stack to create new filter object (c. 6, ll. 51-58; c.

8, ll. 25-37).

Claim Objections

13.   Claims 1, 2, 22, 30-32, 45-47, 58, 63, 74, and 75 are objected to because of the following

informalities: These claims all contain steps that begin with a capital letter. According to MPEP

608.01(m), each claim begins with a capital letter and ends with a period. So only the beginning

of each claim should begin with a capital letter. Appropriate correction is required.

Claim Rejections - 35 USC § 101

14.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or
> any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and
> requirements of this title.

15.   Claims 79-85 are rejected under 35 U.S.C. 101 because the claimed invention is directed

to non-statutory subject matter.

16.   Claims 79-83 are directed to image processing application program interface embodied

on one or more computer readable media. Specification does not appear to explicitly define

"computer readable media". Specification defines application program interface as a high-level

programming language (p. 9, [0047]). Since specification does not define "computer readable

media", one of ordinary skill in the art would fairly determine image processing application

program interface (high-level programming language) could be embodied on "computer readable

media" that could include signals or other forms of propagation and transmission media in order

to transmit image processing application program interface (high-level programming language).

Signals or other forms of propagation and transmission media fail to be appropriate manufacture

under 35 U.S.C. 101 in context of computer-related inventions, and thus "computer readable

media" is directed to non-statutory subject matter (see MPEP 2106 IV).

17.   Claim 84 is directed to application program interface. Specification defines API as high-

level programming language (p. 9, [0047]). Functional descriptive material consists of computer

programs. Descriptive material is nonstatutory when claimed as descriptive material per se (see

MPEP 2106.01). So, API is directed to non-statutory subject matter.

18.   Claim 85 is directed to computer-readable medium having computer executable

instructions. Specification does not appear to explicitly define "computer-readable medium". So,

one of ordinary skill in the art would fairly determine "computer-readable medium" could

include signals or other forms of propagation and transmission media in order to transmit the

instructions. Signals or other forms of propagation and transmission media fail to be appropriate

manufacture under 35 U.S.C. 101 in context of computer-related inventions, and thus "computer-

readable medium" is directed to non-statutory subject matter (see MPEP 2106 IV).

Claim Rejections - 35 USC § 102

19.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on
> sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the
> United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by
> another filed in the United States before the invention by the applicant for patent, except that an international
> application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an
> application filed in the United States only if the international application designated the United States and was
> published under Article 21(2) of such treaty in the English language.

20.   Claims 28, 29, 38, 39, 43, 44, 53, 54, and 74 are rejected under 35 U.S.C. 102(b) as being

anticipated by Grantham (US006215495B1).

21.   As per Claim 28, Grantham teaches processor 108 initiates request which is routed to

server 101. When server 101 receives such request, processor 102 retrieves appropriate VRML

file. VRML file instructs API 112 to make a number of function calls to various graphics engines

113 for performing desired functions on persistent data objects 118. API is structured as collection of class hierarchies (c. 4, ll. 63-c. 5, ll. 23), including Graphics State classes that includes Context, Appearance, Material, Texture, and TexTranform classes (c. 28, ll. 14-16). Context class maintains graphics state for particular graphics context (c. 3, ll. 2-4). The other Graphics State classes define how result image is to be created (c. 8, ll. 38-46). DrawAction class is used to draw scene (c. 7, ll. 31-33). So Grantham teaches method of creating result image having 1st process requesting creation of context; 1st process requesting creation of result image; 1st process indicating parameters associated with creation of result image; 1st process requesting result image be rendered to context (c. 4, ll. 63-c. 5, ll. 23; c. 28, ll. 14-16; c. 3, ll. 2-24; c. 8, ll. 38-46; c. 7, ll. 31-33); 2nd process servicing requests of 1st process, servicing comprising steps of optimizing graph representing result image; compiling optimized graph; causing rendering of compiled optimized graph into context (c. 5, ll. 2-23; c. 4, ll. 55-58; c. 7, ll. 50-57; c. 3, ll. 2-4).

22.   As per Claim 29, Grantham teaches creation of result image comprises editing pre-existing image (c. 11, ll. 22-25).

23.   As per Claim 38, Grantham teaches 1st process is application program (112) (c. 5, ll. 2-5).

24.   As per Claim 39, Grantham teaches second process comprises suite of graphics services functions (c. 3, ll. 1-16; c. 5, ll. 2-19).

25.   As per Claim 43, it is similar in scope to Claim 28, except it has additional limitation that 1st process and 2nd process are running on first microprocessor, and rendering occurs on second microprocessor. Grantham teaches first process (requesting) and second process (compiling) are running on first microprocessor (108), and rendering occurs on second microprocessor (109) (c. 4, ll. 63-c. 5, ll. 23; c. 7, ll. 50-57). So Claim 43 is rejected under same rationale as Claim 28.

26.   As per Claims 44, 53, and 54, these claims are similar in scope to Claims 29, 38, and 39

respectively, and therefore are rejected under the same rationale.

27.   As per Claim 74, Grantham teaches rendering image, 1st process running on CPU 107

requesting creation of image; graphics services resource 109, responding to request by running

1st routine on CPU, 1st routine for optimizing graph representing image; 1st process requesting

rendering of image to specified context; graphics services resource causing rendering of graph

representing image, rendering occurring on GPU (c. 4, ll. 55-58, 63-67; c. 5, ll. 1-23, 61-67).

28.   Thus, it reasonably appears that Grantham describes or discloses every element of Claims

28, 29, 38, 39, 43, 44, 53, 54, and 74 and therefore anticipates the claims subject.

29.   Claims 75 and 78 are rejected under 35 U.S.C. 102(e) as being anticipated by Boudier

(US006995765B2).

30.   As per Claim 75, Boudier teaches converting 1st image representation into 2nd image

representation, comprising creating 1st graph associated with 1st image representation (c. 1, ll. 29-

36; c. 2, ll. 31-34; c. 3, ll. 61-63) where software routines for creating such graph execute on

CPU (504) (c. 1, ll. 49-51; c. 6, ll. 34-38), determining intersection of 1st graph's global domain

of definition (input scene graph) and global region of interest (bounding box) (c. 9, ll. 12-21);

resolving 1st node in 1st graphic by running software routines on CPU to determine if 1st node

may be combined with 2nd node (c. 9, ll. 40-53; c. 5, ll. 33-37) and create program steps for

calculating and storing only portions of any intermediary image that relate to intersection (c. 9, ll.

12-21), program steps for compilation on CPU, execution on GPU (c. 5, ll. 33-37; c. 6, ll. 19-21).

31.   As per Claim 78, Boudier teaches first node is root node (c. 8, ll. 10-11).

32.   Thus, it reasonably appears that Boudier describes or discloses every element of Claims

75 and 78 and therefore anticipates the claim subject.

33.   Claims 79-84 are rejected under 35 U.S.C. 102(e) as being anticipated by McCrossin

(US006600840B1).

34.   As per Claim 79, McCrossin teaches application calls transformation object using API

call (c. 7, ll. 33-35). Transform object 103 determines actual image vector 139, which describes

format of image to be read or written (c. 6, ll. 51-53). Since this describes format of image, this is

related to image objects and context objects. Image request vector is received from application

and is compared to actual image vector 139 to determine what filters are needed (c. 6, ll. 53-56).

Filters are accessed by transformation object from filter library. Filter library contains number of

different filters available for performing various image transformations (c. 6, ll. 59-62). Pointer

to filter vector in filter object 150 points to filter vector 154, which in depicted example is crop

filter (c. 7, ll. 19-21). So, McCrossin teaches image processing API and API services related to

filter objects, API services related to image objects, API services related to context objects and

API services related to vector objects. McCrossin uses API to call at filters. So, McCrossin

teaches image processing application program interface embodied on one or more computer

readable media (c. 6, ll. 25-30), having 1st group of services related to filter objects (c. 6, ll. 59-

67); 2nd group of service related to image objects; 3rd group of services related to context objects

(c. 5, ll. 59-60; c. 6, ll. 51-56); 4th group of services related to vector objects (c. 7, ll. 14-17).

35.   As per Claim 80, McCrossin teaches 1st group of services have 1st functions to create

filter objects; 2nd functions to set filter object parameters; 3rd functions to obtain filter object

output (c. 2, ll. 25-32).

36.   As per Claim 81, McCrossin teaches 2nd group of services have 1st function to create

image objects (c. 6, ll. 51-56); 2nd functions to render an image object to context object (c. 9, ll.

7-18).

37.   As per Claim 82, McCrossin teaches 3rd group of services have 1st functions to create

context objects (c. 5, ll. 46-67).

38.   As per Claim 83, McCrossin teaches 4th group of services comprise 1st functions to create

vector objects (c. 7, ll. 15-32).

39.   As per Claim 84, McCrossin teaches application calls transformation object using API

call (c. 7, ll. 33-35). Filters are accessed by transformation object from filter library 143. Filter

library contains number of different filters available for performing various image

transformations (c. 6, ll. 59-62). Transform object adds filter to filter stack to create new filter

object (c. 6, ll. 51-58; c. 8, ll. 25-37). According to Applicant's disclosure, image, more

commonly, may be created from another image by applying filter to existing image ([0057], p.

12). So, McCrossin teaches API functions to create image objects; API functions to create filter

objects. Transform object 103 determines actual image vector 139, which describes format of

image to be read or written (c. 6, ll. 51-53). According to Applicant's disclosure, creating context

is derived from tools that allow definition of object in memory ([0055], p. 11). Since actual

image vector describes format of image to be read or written, this allows definition of object in

memory, so is considered to create context objects, so McCrossin teaches API functions to create

context objects. Pointer to filter environment in filter object 150 points to memory 156, which

contains information which may be used by filter vector 154 in converting or transforming image

data (c. 7, ll. 21-24), and so McCrossin does teach API functions to set filter object parameters.

Filters are employed by transformation object 103 to provide necessary transformation of image

data to return image data in form as specified in image request vector 127 (c. 6, ll. 66-c. 7, ll. 2),

and so McCrossin teaches API functions to obtain filter object output; API functions to convert

image objects to context objects. So, McCrossin teaches application program interface for

facilitating image processing (c. 6, ll. 25-30), API having functions to create image objects;

create context objects (c. 6, ll. 51-53); create filter objects; set filter object parameters; obtain

filter object output (c. 2, ll. 25-32); convert image objects into context objects (c. 6, ll. 51-53).

40.   Thus, it reasonably appears that McCrossin describes or discloses every element of

Claims 79-84 and therefore anticipates the claims subject.

Claim Rejections - 35 USC § 103

41.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

42.   The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459

(1966), that are applied for establishing a background for determining obviousness under 35

U.S.C. 103(a) are summarized as follows:

    1.   Determining the scope and contents of the prior art.
    2.   Ascertaining the differences between the prior art and the claims at issue.
    3.   Resolving the level of ordinary skill in the pertinent art.
    4.   Considering objective evidence present in the application indicating obviousness
           or nonobviousness.

43.   Claims 1, 2, 8, 11, 12, 14-20, 36, 37, 40, 41, 51, 52, 55, 56, 58, 61-63, 66-73, and 85 are

rejected under 35 U.S.C. 103(a) as being unpatentable over Grantham (US006215495B1) in

view of McCrossin (US006600840B1).

44.   As per Claim 1, Grantham teaches processor 108 initiates request which is routed to

server 101. When server 101 receives such request, processor 102 retrieves appropriate VRML

file. VRML file instructs API 112 to make a number of function calls to various graphics engines

113 for performing desired functions on persistent data objects 118. Interpreter 111 modifies

scene graph to suit specific graphics subsystem hardware 109. Hence, interpreter 111 enables

VRML file to be adapted to run on graphics subsystem hardware (c. 4, ll. 63-c. 5, ll. 11). API is

structured as collection of class hierarchies. Nodes and graphics states are assembled into

cohesive scene graph (c. 5, ll. 18-23). Graphics State classes include Texture class 504 that

defines how image is filtered (c. 28, ll. 10-16; c. 11, ll. 22-25). Outside Scene Graph classes

include CompileAction class 403 that compiles specified subgraph into data structure which is

more efficient for traversals (c. 28, ll. 25-27, 29-30; c. 7, ll. 50-57). So filter is part of scene

graph (c. 5, ll. 18-23; c. 28, ll. 10-16; c. 11, ll. 22-25) that is run on graphics subsystem hardware

(c. 5, ll. 5-11). Before scene graph is run on graphics subsystem hardware (c. 5, ll. 5-11),

CompileAction class compiles it into data structure which is more efficient for traversals (c. 7, ll.

50-57), so compiled data structure includes filter. So VRML file requested by processor 108 (c.

4, ll. 63-c. 5, ll. 11) includes filter (c. 5, ll. 18-23; c. 28, ll. 10-16; c. 11, ll. 22-25), and processor

108 requests filter from compiling process since program needs to be compiled before it is run on

graphics system hardware. So Grantham teaches method of editing initial image, having steps of

1st process requesting (c. 4, ll. 63-c. 5, ll. 11) filter from 2nd process; related filter and initial

image comprising program, second process compiling program, yielding compiled program;

running at least portion of compiled program to apply function of filter to image (c. 5, ll. 5-11,

18-23; c. 28, ll. 10-16; c. 11, ll. 22-25; c. 7, ll. 50-57), yielding pixel-image result (c. 5, ll. 9-11).

However, Grantham does not teach 1st process defines relationship between filter and

initial image. But, McCrossin teaches editing initial image, 1st process requesting filter from 2nd

process; 1st process defining relationship between filter and initial image (c. 6, ll. 46-c. 7, ll. 9).

It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify Grantham so first process defines relationship between filter and initial

image because McCrossin suggests reducing amount of processing required to convert image

data from original or input format into desired output format (c. 1, ll. 54-57; c. 2, ll. 3-24).

45.   As per Claim 2, Grantham teaches optimizing the program (c. 4, ll. 55-58).

46.   As per Claim 8, Grantham teaches compiled program has component for 1st (108, Fig. 1)

and 2nd microprocessors (109) (c. 4, ll. 63-c. 5, ll. 11; c. 7, ll. 50-57).

47.   As per Claim 11, Grantham teaches first microprocessor is CPU (108, Fig. 1) and second

microprocessor is GPU (109) (c. 4, ll. 63-c. 5, ll. 11).

48.   As per Claim 12, Grantham teaches first and second microprocessors are both CPUs

(108, 102, Fig. 1; c. 4, ll. 63-c. 5, ll. 11).

49.   As per Claim 14, Grantham teaches initial image is only color (c. 12, ll. 45-52).

50.   As per Claim 15, Grantham teaches 1st process is application program (112) (c. 5, ll. 2-5).

51.   As per Claim 16, Grantham teaches second process comprises suite of graphics services

functions (c. 3, ll. 1-16; c. 5, ll. 2-19).

52.   As per Claim 17, Grantham does not explicitly teach operating system comprises second

process. However, McCrossin teaches this limitation (c. 4, ll. 2-5).

It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify device of Grantham so operating system comprises second process because

McCrossin suggests operating system is needed in order for processor to function (c. 4, ll. 2-5).

53.   As per Claim 18, Grantham teaches VRML is high level programming language (c. 1, ll.

63-67). Processor 108 initiates request which is routed to server 101. When server 101 receives

such request, processor 102 retrieves appropriate VRML file. VRML file instructs API 112 to

make a number of function calls to various graphics engines 113 for performing desired

functions on persistent data objects 118. Interpreter 111 modifies scene graph to suit specific

graphics subsystem hardware 109. Hence, interpreter 111 enables VRML file to be adapted to

run on graphics subsystem hardware (c. 4, ll. 63-c. 5, ll. 11). API is structured as collection of

class hierarchies. Nodes and graphics states are assembled into cohesive scene graph (c. 5, ll. 18-

23). Graphics State classes include Texture class 504 that defines how image is filtered (c. 28, ll.

10-16; c. 11, ll. 22-25). So high level programming language VRML (c. 1, ll. 63-67) is

interpreted by interpreter 111 into lower level programming language so that it is adapted to run

on graphics subsystem hardware (c. 5, ll. 5-11), and this lower level programming language

includes filter (c. 5, ll. 18-23; c. 28, ll. 10-16; c. 11, ll. 22-25). So first process requests filter by

requesting VRML file (c. 4, ll. 63-67), which is high level programming language (c. 1, ll. 63-

67), and second process responds by interpreting high level programming language into lower

level programming language (c. 5, ll. 5-11) including filter so that filtering is adapted to run on

graphics subsystem hardware (c. 5, ll. 18-23; c. 28, ll. 10-16; c. 11, ll. 22-25). So first process

requests high-level filter (c. 4, ll. 63-67; c. 1, ll. 63-67) and second process responds with object

representing lower-level filter (c. 5, ll. 5-11, 18-23; c. 28, ll. 10-16; c. 11, ll. 22-25).

54.   As per Claim 19, Grantham shows processor 108, interpreter 111, and API 112 are part of

computer system 116 in Fig. 1. So first process and second process run on CPU (108, 111, 112)

and compiled program runs on GPU (109) (c. 4, ll. 63-c. 5, ll. 23; c. 7, ll. 50-57).

55.   As per Claim 20, it is similar in scope to Claim 19, so is rejected under same rationale.

56.   As per Claim 36, Grantham teaches first process requests output of graph

programmatically assembled by first process (c. 4, ll. 63-c. 5, ll. 23).

However, Grantham does not explicitly teach graph has one or more pre-defined filters.

But, McCrossin teaches this (c. 6, ll. 59-67). This would be obvious for reasons for Claim 1.

57.   As per Claim 37, Grantham teaches first process programmatically assembled graph in

cooperation with second process (compiling) (c. 4, ll. 63-c. 5, ll. 23; c. 7, ll. 50-57).

58.   As per Claims 40 and 41, these claims are each similar in scope to Claim 17, and so are

rejected under the same rationale. As per Claims 51, 52, 55, and 56, these claims are similar in

scope to Claims 36, 37, 40, and 41 respectively, and so are rejected under the same rationale.

59.   As per Claim 58, Grantham teaches interpreter 111 enables VRML file to be adapted to

run on graphics system hardware 109 (c. 5, ll. 5-9). VRML is high level programming language

(c. 1, ll. 63-67). So Grantham teaches providing high level interface (111) to graphics processing

resource (109) (c. 5, ll. 5-9; c. 1, ll. 63-67) having 1st process requesting performance of task

through 1 or more function calls serviced by graphics processing resource (c. 4, ll. 63-c. 5, ll.

23), function calls selected from 1 or more of following options, (i) creating of image (Graphics

State classes) (c. 28, ll. 14-16; c. 8, ll. 37-46; c. 7, ll. 31-33), (iv) asking for output of filter or

another filter (504; c. 11, ll. 22-25), (v) creating context (302); (vi) rendering image to context or another context; request having object associated therewith, object comprising one of following: context (c. 5, ll. 61-67), image (c. 28, ll. 14-16; c. 8, ll. 37-46; c. 7, ll. 31-33), filter (c. 11, ll. 22-25), or vector (c. 26, ll. 20-22); graphics processing resource servicing request (c. 5, ll. 2-23).

However, Grantham does not explicitly teach request defines relationship between at least one of functions and one of objects. However, McCrossin teaches this limitation (c. 6, ll. 46-c. 7, ll. 9). This would be obvious for the same reasons given in the rejection for Claim 1.

60. As per Claim 61, Grantham teaches creating graph representing image (c. 2, ll. 52-58).

61. As per Claim 62, Grantham teaches optimizing graph (c. 4, ll. 55-59; c. 5, ll. 2-11).

62. As per Claims 63, 66, and 67, these claims are similar in scope to Claims 58, 61, and 62 respectively, and so are rejected under same rationale.

63. As per Claim 68, Grantham does not explicitly teach function calls include option of creating filter. But, McCrossin teaches this (c. 6, ll. 59-67). This is for reasons for Claim 1.

64. As per Claim 69, Grantham does not explicitly teach function calls include option of setting arguments associated with filter. However, McCrossin teaches this limitation (c. 7, ll. 1-7; c. 6, ll. 9-21). This would be obvious for the same reasons given in the rejection for Claim 1.

65. As per Claim 70, it is similar in scope to Claim 69, so is rejected under same rationale.

66. As per Claim 71, Grantham does not explicitly teach another object associated with request has filter. But, McCrossin teaches this (c. 6, ll. 59-67). This would be obvious for reasons for Claim 1.

67.   As per Claim 72, Grantham does not explicitly teach another object associated with

request is vector. But, McCrossin teaches this (c. 7, ll. 14-17). This would be obvious for reasons

for Claim 1.

68.   As per Claim 73, it is similar in scope to Claim 72, so is rejected under same rationale.

69.   As per Claim 85, Grantham teaches computer executable instructions for performing

method recited in Claim 1 (c. 4, ll. 63-c.5, ll. 23).

70.   Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Grantham

(US006215495B1) and McCrossin (US006600840B1) in view of Levy (US 20020033844A1).

        Grantham and McCrossin are relied upon for teachings for Claim 2.

        But, Grantham and McCrossin do not teach optimizing includes step of using cache look-

up to see if pixel-image result is already in cache. But, Levy teaches this [0184, 0021, 0038].

        It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify devices of Grantham and McCrossin so step of optimizing includes step of

using a cache look-up to see if the pixel-image result is already in cache because Levy suggests

advantage of avoiding repeated read operations on same content [0184].

71.   Claims 4-6 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Grantham (US006215495B1) and McCrossin (US006600840B1) in view of Boudier

(US006995765B2).

72.   As per Claim 4, Grantham and McCrossin are relied on for teachings relative Claim 2.

        But, Grantham and McCrossin do not teach optimizing includes step of calculating

intersection, intersection representing area where pixel-image result is both defined and in result

region requested of 2nd process. But, Boudier teaches step of optimizing includes step of

calculating intersection, intersection representing area where pixel-image result is both defined (input scene graph) and in result region requested of 2nd process (bounding box) (c. 9, ll. 12-21).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify devices of Grantham and McCrossin so step of optimizing includes step of calculating intersection, intersection representing area where pixel-image result is both defined and in result region requested of second process because Boudier suggests advantage of removing unnecessary regions, which reduces traversal time (c. 9, ll. 12-21).

73.   As per Claim 5, Grantham does not teach step of using calculated intersection to limit number of pixels that require calculation during running of compiled program. However, Boudier teaches step of using calculated intersection (intersection of input scene graph and bounding box) to limit number of pixels that require calculation during running of compiled program (c. 9, ll. 12-21; c. 8, ll. 52-53). This would be obvious for reasons for Claim 4.

74.   As per Claim 6, Grantham doesn't teach using calculated intersection to limit amount of memory necessary for storing calculated images. Boudier teaches memory usage is increased during actual calculating of intersection due to creation of bounding volumes. But, after creation of bounding boxes, after the intersection is calculated, the unnecessary bounding boxes are removed. Bounding box is used for view frustum culling algorithm, so data outside of bounding box is culled out, so only data inside bounding box needs to be stored, so calculated intersection (intersection of input scene graph and bounding box) is used to limit amount of memory necessary for storing calculated images (c. 9, ll. 12-21). This is obvious for reasons for Claim 4.

75.   As per Claim 21, it is similar in scope to Claim 19, so is rejected under same rationale.

76. Claims 7, 9, 60, and 65 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Grantham (US006215495B1) and McCrossin (US006600840B1) in view of Parikh

(US006411301B1).

77. As per Claim 7, Grantham and McCrossin are relied on for teachings relative to Claim 1.

However, Grantham and McCrossin do not teach compiled program is for single

microprocessor. However, Parikh teaches single microprocessor that emulates graphics

processing (c. 26, ll. 29-62). So, Grantham can be modified so only single microprocessor is used

instead of using separate graphics processor (c. 5, ll. 2-23) so compiled program of Grantham (c.

7, ll. 50-57) is for single microprocessor, as suggested by Parikh.

It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify Grantham and McCrossin so compiled program is for single microprocessor

because Parikh suggests advantage of still being able to perform graphics processing even when

graphics processor for which software is written for is not available (c. 26, ll. 29-62).

78. As per Claim 9, Grantham does not teach single microprocessor is CPU. However, Parikh

teaches this limitation (c. 26, ll. 29-62). This would be obvious for reasons given for Claim 7.

79. As per Claim 60, Grantham does not teach request is serviced using emulator to run GPU

program on CPU. But, Parikh teaches this (c. 26, ll. 29-62). This would be obvious for reasons

for Claim 7.

80. As per Claim 65, it is similar in scope to Claim 60, so is rejected under same rationale.

81. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Grantham

(US006215495B1), McCrossin (US006600840B1), and Parikh (US006411301B1) in view of

Doyle (US006867779B1).

Grantham, McCrossin, and Parikh are relied on for teachings relative to Claim 7.

However, Grantham, McCrossin, and Parikh do not teach single microprocessor is programmable GPU. However, Doyle teaches dividing up rendering between CPU and GPU based on progress of one device of two devices (c. 1, ll. 20-26; c. 2, ll. 1-3). So, if CPU is busy, rendering is only performed in GPU. So, single microprocessor is programmable GPU.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify devices of Grantham, McCrossin, and Parikh so single microprocessor is programmable GPU because Doyle suggests advantage of using device that would most quickly process command to process command (c. 1, ll. 20-26).

82.   Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Grantham (US006215495B1) and McCrossin (US006600840B1) in view of Nelson (US006801202B2).

Grantham and McCrossin are relied on for teachings for Claim 8.

But, Grantham and McCrossin do not teach 1st and 2nd microprocessors are both GPUs. But, Nelson teaches API to generate commands and graphics data, and these commands are processed by 1st and second microprocessors that are both GPUs (c. 8, ll. 32-42, 67; c. 9, ll. 1-3).

It would have been obvious to one of ordinary skill in the art to modify devices of Grantham and McCrossin so 1st and 2nd microprocessors are both GPUs because Nelson suggests advantage of increased graphics performance (c. 3, ll. 47-c. 63).

83.   Claims 22, 23, and 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Grantham (US006215495B1) and McCrossin (US006600840B1) in view of Hoppe (US006919906B2).

84.   As per Claim 22, Grantham teaches system for editing initial image, comprising first

microprocessor (108, 111, 112, Fig. 1) running first process and second process (compiling) for

servicing requests from first process (c. 4, ll. 63-c. 5, ll. 11; c. 7, ll. 50-57). API is structured as a

collection of class hierarchies (c. 5, ll. 16-23), and these classes include Texture class 504 that

defines how image is filtered (c. 11, ll. 22-25). So filter is in API (c. 5, ll. 16-23; c. 11, ll. 22-25),

and API 112 is stored in memory 110, as shown in Fig. 1. So memory 110 is for storing filter,

second memory for storing data structure comprising information used in first process (c. 5, ll.

16-23; c. 11, ll. 22-25; Fig. 1) (first and second memories are the same, as recited in Claim 23);

second microprocessor (109) for running program created using data structure; outputting pixel

image resulting from running program (c. 5, ll. 2-11).

        But, Grantham doesn't teach data structure has relationship between initial image and

filter. But, McCrossin teaches data structure 103 has relationship between initial image and

specific filter (*image request vector 127 is received from application 101 and is compared to*

actual image vector to determine what filters are needed, c. 6, ll. 46-c. 7, ll. 9; *application calls*

the transformation object using API call, c. 7, ll. 33-35). This is obvious for reasons for Claim 1.

        However, Grantham and McCrossin do not explicitly teach third memory for storing

pixel image resulting from running program. However, Hoppe teaches API (c. 2, ll. 20-26) for

filtering (c. 1, ll. 34-36), and third memory (208) for storing pixel image resulting from running

program (c. 4, ll. 36-37; c. 9, ll. 33-36; c. 1, ll. 55-58).

        It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify Grantham and McCrossin to include 3rd memory for storing pixel image

resulting from running program because Hoppe suggests it is well-known in the art to have frame

buffer in video memory to store rendered image so that rendered image can be output from video memory to display (c. 4, ll. 36-37; c. 9, ll. 33-36).

85.   As per Claim 23, Grantham teaches first and second memories are the same (Fig. 1; c. 4, ll. 18-21; c. 5, ll. 2-23).

86.   As per Claim 25, Grantham teaches first and second memories are in system memory (110, Fig. 1; c. 4, ll. 18-21; c. 5, ll. 2-23)

    However, Grantham does not teach third memory is in video memory. However, Hoppe teaches this limitation (208; c. 4, ll. 36-37). This would be obvious for reasons for Claim 22.

87.   As per Claim 26, Grantham teaches first microprocessor (108, Fig. 1) processes data structure to produce program for use on second microprocessor (109) (c. 4, ll. 63-c. 5, ll. 11).

88.   As per Claim 27, Grantham teaches second microprocessor (109, Fig. 1), under control of program causes pixel image to be output (c. 5, ll. 2-11).

    But, Grantham doesn't teach image is stored in 3rd memory. But, Hoppe teaches 2nd processor 206, under control of program causes pixel image to be stored in 3rd memory 208 (c. 4, ll. 36-44; c. 9, ll. 33-36; c. 1, ll. 55-58; c. 2, ll. 20-26). This is obvious for reasons for Claim 22.

89.   Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Grantham (US006215495B1), McCrossin (US006600840B1), and Hoppe (US006919906B2) in view of Sturges (US005854637A).

    Grantham and McCrossin are relied on for teachings relative to Claim 22. Grantham teaches first and second memories are system memory (Fig. 1; c. 4, ll. 18-21; c. 5, ll. 2-23). Hoppe teaches third memory is frame buffer (208; c. 4, ll. 36-37), as discussed for Claim 22.

But, Grantham, McCrossin do not teach 1st, 2nd, 3rd memories are the same. But, Sturges

teaches using shared memory for both system memory and frame buffer (c. 3, ll. 7-15). So

combination of Grantham, McCrossin can be modified so 1st and 2nd memories in system

memory of Grantham, frame buffer of Hoppe are in the same memory as suggested by Sturges.

It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify devices of Grantham and McCrossin so first, second and third memories are

the same because Sturges suggests advantage of being able to use unused portions of frame

buffer memory to be employed as system memory (c. 1, ll. 35-41).

90.   Claims 30-32 and 45-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Grantham (US006215495B1) in view of Boudier (US006995765B2).

91.   As per Claim 30, it is similar in scope to Claim 4, and so is rejected under same rationale.

92.   As per Claim 31, Grantham does not teach step of optimizing graph representing first

image comprises step of analyzing adjacent nodes in graph for purpose of attempting to

consolidate nodes. However, Boudier teaches this (c. 9, ll. 40-53; c. 8, ll. 4-10).

It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify device of Grantham so step of optimizing graph representing first image

comprises step of analyzing adjacent nodes in graph for purpose of attempting to consolidate

nodes because Boudier teaches this reduces the number of nodes, so reduces memory usage, and

drawing time is reduced because there will be fewer function calls (c. 8, ll. 20-24; c. 9, ll. 40-53).

93.   As per Claim 32, Claim 32 is similar in scope to Claim 31, and so is rejected under same

rationale. As per Claims 45-47, these claims are similar in scope to Claims 30-32 respectively,

and so are rejected under same rationale.

94.   Claims 33 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Grantham (US006215495B1) and Boudier (US006995765B2) in view of Levy (US

20020033844A1).

Grantham and Boudier are relied upon for teachings as discussed relative to Claim 31.

However, Grantham does not teach explicitly teach step of analyzing adjacent nodes and

storing result of such analysis in memory. However, Boudier teaches this (c. 9, ll. 40-53; c. 8, ll.

20-21). This would be obvious for reasons for Claim 31.

But, Grantham and Boudier do not teach checking cache to determine if result of such

analysis is available in memory. But, Levy teaches step of checking cache to determine if pixel-

image result is available in memory [0184, 0021, 0038]. This is obvious for reasons for Claim 3.

95.   Claims 34, 35, 49, and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Grantham (US006215495B1) in view of Levy (US 20020033844A1).

96.   As per Claim 34, Grantham is relied on for teachings for Claim 28. Grantham teaches

step of optimizing graph and storing optimized graph in memory (c. 4, ll. 55-58; c. 2, ll. 54-67).

However, Grantham does not teach step of checking cache to determine if graph has

already been optimized. But, Levy teaches step of checking cache to determine if pixel-image

result is already in memory [0184, 0021, 0038]. This would be obvious for reasons for Claim 3.

97.   As per Claim 35, Grantham teaches step of optimizing graph and outputting result of

rendering graph (c. 4, ll. 55-58; c. 5, ll. 2-23).

But, Grantham does not teach step of checking cache to determine if result of rendering

graph is available in memory. But, Levy teaches step of checking cache to determine if pixel-

image result is available in memory [0184, 0021, 0038]. This is obvious for reasons for Claim 3.

98.   As per Claims 49 and 50, these claims are similar in scope to Claims 34 and 35

respectively, and so are rejected under same rationale.

99.   Claims 42 and 57 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Grantham (US006215495B1) in view of Stokes (US006977661B1).

        Grantham is relied on for teachings for Claim 28. Grantham teaches 2nd process inserts

nodes into graph (c. 5, ll. 2-23) for performing certain functions (c. 4, ll. 23-49).

        However, Grantham does not teach functions include functions for converting original

color scheme to operating color scheme and for converting operating color scheme back to

original color scheme. However, Stokes teaches this limitation (c. 5, ll. 63-67).

        It would have been obvious to one of ordinary skill in the art at the time of invention by

applicant to modify Grantham so 2nd process inserts nodes into graph for converting original

color scheme to operating color scheme and for converting operating color scheme back to

original color scheme because Stokes suggests this is needed because color data generated by

image-capturing device are generally in device-specific color space that is different from color

space or spaces used by image-processing application for image editing (c. 1, ll. 40-44).

100.   Claims 59 and 64 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Grantham (US006215495B1) and McCrossin (US006600840B1) in view of Doyle

(US00687779B1).

        Grantham and McCrossin are relied on for teachings discussed relative to Claim 58.

        However, Grantham and McCrossin do not teach request is serviced using GPU and

CPU. However, Doyle teaches request is serviced using GPU and CPU (c. 2, ll. 1-3).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Grantham and McCrossin so request is serviced using GPU and CPU because Doyle teaches this accelerates speed by which 3-D image can be rendered (c. 1, ll. 7-10).

101. Claims 76 and 77 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boudier (US006995765B2) in view of Levy (US 20020033844A1).

102. As per Claim 76, Boudier is relied on for teachings for Claim 75. Boudier teaches step of determining if first node may be combined with second node, and result is stored in memory (c. 9, ll. 40-53; c. 8, ll. 20-21).

However, Boudier does not teach step of checking cache to determine if there is result is already in memory regarding such determination regarding combining nodes. However, Levy teaches step of checking cache to determine if pixel-image result is already in memory [0184, 0021, 0038]. This would be obvious for the same reasons given in the rejection for Claim 3.

103. As per Claim 77, Boudier teaches step of resolving first node and storing resolution of first node in memory (c. 9, ll. 40-53; c. 8, ll. 20-21).

However, Boudier does not teach step of resolving first node comprises step of checking cache to determine if there is result in memory regarding resolution of first node. However, Levy teaches step of checking cache to determine if pixel-image result is already in memory [0184, 0021, 0038]. This would be obvious for the same reasons given in the rejection for Claim 3.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joni Hsu whose telephone number is 571-272-7785. The examiner can normally be reached on M-F 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Kee Tung can be reached on 571-272-7794. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would

like assistance from a USPTO Customer Service Representative or access to the automated

information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JH

KEE M. TUNG
SUPERVISORY PATENT EXAMINER